

REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application in view of the foregoing amendments and the following remarks.

Claims 1-9, 11-15, 17-22, 24-34, 36-43, and 45-50 were previously submitted. Claims 1-9, 11-15, 17-22, 24-34, 36-43, and 45 remain pending in the application, with claims 1, 12, 19, 26, and 38 being independent. Applicant amends independent claims 1, 12, 19, 26, and 38 to include the elements of canceled dependent claims 45-50, respectively. Applicant amends dependent claim 14 for consistency with independent claim 12. Applicant adds new claim 51. No new matter has been added.

Applicant cancels claims 45-50 without prejudice, waiver, or disclaimer of the subject matter.

Applicant thanks the Office for withdrawing the previous §101 and §112 rejections.

§102 REJECTIONS

Claims 1-9, 11-15, 17-22, 24-34, 36-43, and 45-50 stand rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,915,085 (Koved). Applicant respectfully traverses the rejection.

The MPEP states that “[a] claim is anticipated **only if each and every element as set forth in the claim is found**, either expressly or inherently described, in a single prior art reference. . . . The identical invention must be shown in as complete detail as is contained in the . . . claim. . . . The elements must be arranged as required by the claim” MPEP §2131 (emphasis added). Consequently, under the guidelines of the MPEP set

forth above, if there is *any* substantial difference between the prior art cited by the Office and an applicant's claim, the prior art does NOT establish a *prima facie* case of anticipation and, barring other rejections, such claim is allowable over the cited prior art.

Nevertheless, without conceding the propriety of the rejection and only in the interest of expediting allowance of the application, Applicant amends, **independent claim 1**. Support for this amendment is found at least in dependent claim 46 and paragraph 0004 of the application. Thus no new matter has been added.

Independent claim 1, as amended, recites a method:

to estimate security requirements needed to execute managed code comprising:

simulating the execution of all execution paths of one or more assemblies in managed code, *wherein an assembly comprises one or more files versioned and deployed as a unit, wherein the managed code is a managed shared library or an executable, wherein all managed code is contained within the one or more assemblies*; and

finding a set of required permissions for each execution path by one or more simulated stack walks that each include a plurality of the assemblies, wherein each call in each execution path has a corresponding permissions set, and wherein *the simulated stack walk comprises:*

entering a public entry point of a method in the assembly;

gathering a permission set for the method;

determining whether the method calls another method;

gathering a permission set for the called method;

and

creating a union of the gathered permission sets.

Applicant respectfully submits that no such method is disclosed by Koved.

Koved is directed to automatically determining “access rights required by Java programs or libraries ... [using] a modified interprocedural invocation graph, called an

access rights invocation graph (ARIG), to compute the access rights” (page 2, column 1, paragraph 1). Koved describes the computation of those access rights by an approach in which “[f]or each node $n \in N$, the algorithm determines $RP(n)$ by starting from $RP(N_{stop})$, and tracing paths back from nodes in N_{stop} to nodes in N_{start} ” (page 3, column 2, paragraph 3). In Koved, “ $N_{stop} := N_{cp}$ ” and “ $N_{cp} := \{n(M,R,P) | M \text{ is } AccessController.checkPermission\}$ ” (page 3, column 2, paragraph 1). The approach starts “with estimates for $RP(s)$ for every s ” (page 3, column 2, paragraph 5) where “ $RP(N)$ [is] the required Permissions for the nodes in N ” (page 3, column 2, paragraph 2) and s is another node on a path from node n (page 3, column 2, paragraph 4). The approach next works backwards (page 3, column 2, paragraph 5) along the “inward adjacencies of n ” (page 3, column 1, paragraph 5) “towards nodes in N_{start} ” (page 3, column 2, paragraph 5). “Finally, $RP(C)$, the set of Permissions required for a class C can then be computed” as the union of the required Permissions for each node n (page 3, column 2, paragraph 6). As agreed during the interview, Koved does not disclose that “*the simulated stack walk comprises: entering a public entry point of a method in the assembly,*” as recited in claim 1.

Additionally, Applicant has searched and failed to find any disclosure in Koved of “*an assembly [that] comprises one or more files versioned and deployed as a unit, wherein the managed code is a managed shared library or an executable, wherein all managed code is contained within the one or more assemblies,*” as recited in amended claim 1. Accordingly, for at least the above reasons each and every element of claim 1 is not found in Koved.

Without conceding the propriety of the rejection and only in the interest of expediting allowance of the application, Applicant amends, **independent claim 12**. Support for this amendment is found at least in dependent claim 47, paragraph 0023, and Figure 3b of the application. Thus no new matter has been added.

Independent claim 12, as amended, recites in a managed code environment, a method comprising:

- simulating calling from one assembly to another for which a permission set is required, wherein the simulation comprises one or more simulated stack walks that include two or more of the assemblies, *each assembly being managed code in a library*, and wherein *the simulated stack walk comprises:*

- entering a public entry point of a method in the assembly;*

- gathering a permission set for the method;
 - determining whether the method calls another method;

- for each called method:
 - gathering a permission set for the called method;
 - and

- determining whether the called method calls a subsequent method; and

- creating a union of the gathered permission sets;
 - repeating the calling for each assembly in the managed code and for all possible execution paths of the managed code;

- repeating the entering for each public entry point in the library;* and

- finding the union of the permission sets corresponding to each call.

Applicant respectfully submits that no such method is disclosed by Koved.

For the same reasons as presented above with respect to independent claim 1, and as agreed during the interview, Koved does not disclose that “*the simulated stack walk*

comprises: entering a public entry point of a method in the assembly,” as recited in claim 12.

Additionally, Applicant has searched and failed to find any disclosure in Koved of “*each assembly being managed code in a library [and] ... repeating the entering for each public entry point in the library*” as recited in amended claim 12. Accordingly, for at least the above reasons each and every element of claim 12 is not found in Koved.

Without conceding the propriety of the rejection and only in the interest of expediting allowance of the application, Applicant amends, **independent claim 19**. Support for this amendment is found at least in dependent claim 48 and paragraphs 0002-0003 of the application. Thus no new matter has been added.

Independent claim 19, as amended, recites one or more computer-readable storage media

having a tangible component comprising instructions that, when executed, perform a simulation of the execution of every data and control flow for managed code from which an estimate is derived of the minimum security requirements needed to dynamically execute the managed code without triggering a security exception, wherein the simulation of the execution comprises, for each data and control flow for the managed code, one or more simulated stack walks that include two or more of the assemblies, wherein *the managed code makes use of a common language runtime (CLR) that is loaded upon the first invocation of a routine*, and wherein *the simulated stack walk comprises:*

entering a public entry point of a method in the assembly;

gathering a permission set for the method;

determining whether the method calls another method;

for each called method:

gathering a permission set for the called method;
and
determining whether the called method calls a
subsequent method; and
creating a union of the gathered permission sets.

Applicant respectfully submits that no such computer-readable storage media is disclosed by Koved.

For the same reasons as presented above with respect to independent claim 1, and as agreed during the interview, Koved does not disclose that “*the simulated stack walk comprises: entering a public entry point of a method in the assembly,*” as recited in claim 19.

Additionally, Applicant has searched and failed to find any disclosure in Koved of “*the managed code makes use of a common language runtime (CLR) that is loaded upon the first invocation of a routine,*” as recited in amended claim 19. Accordingly, for at least the above reasons each and every element of claim 19 is not found in Koved.

Applicant amends, **independent claim 26** to include the elements formerly in dependent claim 49. Thus, this claim presents matter already examined.

Independent claim 26, as amended, recites an apparatus comprising:

means for processing;
means for storing information in memory coupled
to the means for processing;
virtual machine means, stored in the memory, in a
managed code portion, for operating a plurality of
assemblies in managed code, wherein the managed code is
a managed shared library or an executable and is in the
managed code portion;
execution engine means, in a native code
portion, for executing the virtual machine means;

means, in the native code portion, for providing an operating system;

means for making a call in the managed code portion for access by one assembly to another assembly for which a permissions set is required;

means in the managed code portion for gathering the permissions set from each call;

means in the managed code portion for deriving a union of the gathered permissions sets; and

means in the managed code portion for simulating the execution of all possible execution paths for the managed shared library or the executable to derive therefrom the derived union of the gathered permissions sets wherein the means for simulating the execution performs, for each execution path, one or more simulated stack walks that each include a plurality of assemblies, and wherein the *one or more simulated stack walks comprise:*

means for entering a public entry point of a method in the assembly;

means for gathering a permission set for the method;

means for determining whether the method calls another method;

for each called method:

means for gathering a permission set for the called method;

means for determining whether the called method calls a subsequent method; and

means for repeating the previous gathering and determining until any gathered permission set is duplicative; and

means for creating a union of the gathered permission sets.

Applicant respectfully submits that no such apparatus is disclosed by Koved.

For the same reasons as presented above with respect to independent claim 1, and as agreed during the interview, Koved does not disclose “*one or more simulated stack walks comprise: means for entering a public entry point of a method in the assembly; means for gathering a permission set for the method; means for determining whether the method calls another method; for each called method: means for gathering a permission*

set for the called method; means for determining whether the called method calls a subsequent method; and means for repeating the previous gathering and determining until any gathered permission set is duplicative; and means for creating a union of the gathered permission sets.” Accordingly, Applicant respectfully submits that Koved does not establish a *prima facie* case of anticipation as required by MPEP §2131 because each and every element of claim 26 is not found in Koved.

Applicant amends, **independent claim 38** to include the elements formerly in dependent claim 50. Thus, this claim presents matter already examined.

Independent claim 38, as amended, recites a computing device comprising:

- a processor;
- a memory coupled to the processor;
- a managed code portion stored in the memory including a plurality of assemblies each being managed code in a managed shared library or in an executable;
- a native code portion stored in the memory including:
 - an execution engine; and
 - an operating system under the execution engine;
 - a virtual machine interfaced between the managed code portion and the native code portion and executed by the execution engine;
 - an application program in the managed code portion comprising logic configured to:
 - simulate the execution of all possible calls from one assembly to another for all possible execution paths of the managed code, wherein each assembly call has a corresponding permissions set, wherein the simulation of the execution comprises one or more simulated stack walks that each include a plurality of the assemblies, and wherein ***the one or more simulated stack walks comprise:***
 - a public entry point of a method in the assembly;***
 - a permission set for the method;***
 - a determination of whether the method calls another method;***
 - for each called method;***

*a permission set for the called method;
a determination of whether the called method calls
a subsequent method; and
a totality of permission sets such that any
subsequent permission set is duplicative; and
a union of the permission sets; and
derive a union of the permissions sets from each
assembly call.*

Applicant respectfully submits that no such computing device is disclosed by Koved.

For the same reasons as presented above with respect to independent claim 1, and as agreed during the interview, Koved does not disclose “*one or more simulated stack walks comprise: a public entry point of a method in the assembly; a permission set for the method; a determination of whether the method calls another method; for each called method: a permission set for the called method; a determination of whether the called method calls a subsequent method; and a totality of permission sets such that any subsequent permission set is duplicative; and a union of the permission sets.*” Accordingly, Applicant respectfully submits that Koved does not establish a *prima facie* case of anticipation as required by MPEP §2131 because each and every element of claim 38 is not found in Koved.

Dependent claims 2-9, 11, 13-15, 17-18, 20-22, 24-25, 27-34, 36-37, 39-43, and 45 each depended directly or indirectly from an allowable base claim and are allowable by virtue of that dependency, as well as for additional features that each recites. Applicant also respectfully requests individual consideration of every dependent claim.

Applicant respectfully requests withdrawal of the §102 rejections of claims 1-9, 11-15, 17-22, 24-34, 36-43, and 45.

NEW CLAIM 51

New dependent claim 51 depends from independent claim 12 and recites, in part, *“wherein the union of the permission sets separately identifies a permission set for each public entry point of the library.”* Support for this addition is found at least in paragraph 0023 of the application as originally filed. Applicant respectfully requests allowance of new claim 51 because it depends from an allowable base claim as well as for the addition features it recites.

CONCLUSION

For at least the foregoing reasons, claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 are in condition for allowance. Applicant respectfully requests reconsideration and withdrawal of the rejections and an early notice of allowance. The arguments and amendments presented herein were necessitated by the most recent Office Action, and could not have been presented previously because Applicant earnestly believed that the claims were in condition for allowance at the time of filing the previous response.

If any issue remains unresolved that would prevent allowance of this case, **Applicant requests that the Examiner contact the undersigned representative to resolve the issue before issuing a subsequent Action.**

Respectfully submitted,

Lee & Hayes, PLLC

Date: July 22, 2008

By: /Benjamin Keim/
Benjamin A. Keim
Reg. No. 59,217
(509) 324-9256 x248

David A. Divine
Reg. No. 51,275
(509) 324-9256 x233